**Macintosh HD:Users:SamanthaChang:Desktop:AYLA:Logos:AylaNetworksTransparentsmall.png**

Getting Started with Ayla Mobile Libraries for Android

Getting Started with aAML

Version: 3.49

12/02/2014

Installation

AML is easiest to build using Eclipse Indigo SR2+ and ADT. Links for them and other dependencies are found at the top of the AML documentation.

Installation steps:

1. Create a new Android Eclipse project.
2. - Copy the aAML.jar file to the libs directory of your Android executable project. Or…

- Build AML by installing the AML Android source code outside of your Eclipse workspace by unzipping the Android\_AylaLibrary-vXXX.zip. To build AML use File/Import/Android/Existing Android Code Into Workspace in Eclipse. Add the required external libraries to the AML libs directory. Refresh, Clean, and Build the project. File/Export/Java/JAR file, the aAML.jar file unselecting non-required files like the manifest and documentation.

1. Download the AML dependency jar files (gson, log4j, etc.) and add them to the new Eclipse project in a new “libs” directory.
2. Use Eclipse “package explorer”/”build path”/”Configure Build Path…” and add the jars to the class path as necessary.
3. At the top of your main activity file, add:
   * import com.aylanetworks.aaml.\*;
4. After onCreate() add these lines:
   * AylaNetworks aylaNetworks = new AylaNetworks(this, gblAmlDeviceSsidRegex, appId);
   * Where you replace gblAmlDeviceSsidRegex with a regular expression string matching the SSID of your company’s device when they are in AP mode.
   * This is required for the Setup Task to work. Set gblAmlDeviceSsidRegex to “^Ayla-[0-9A-Fa-f]{12}” for Ayla EVB/Dev Kits.
   * String appId = “aMCA-id”;
   * String appSecret = “aMCA-9097620”;

* Use these values until you receive specific replacements for your company from your Ayla technical representative/Dashboard OEM Administrator.

1. In the project AndroidManifest.xml:
   * Add the following uses and permissions:

<uses-sdk android:minSdkVersion="10" android:targetSdkVersion="15" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.RECEIVE\_SMS" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.BROADCAST\_STICKY" />

<uses-permission android:name="android.permission.CHANGE\_WIFI\_MULTICAST\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_COURSE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

* + Add a <receiver> identification section between the <application …. </application> indicators:

<receiver android:name="com.aylanetworks.aaml.AylaConnectivyListener">

<intent-filter>

<action android:name="android.net.conn.CONNECTIVITY\_CHANGE"/>

</intent-filter>

</receiver>

* + Add a <service> identification line between the <application …. </application> indicators:

<service android:name="com.aylanetworks.aaml.AylaExecuteRequest"></service>

For an example of how this should look when complete, refer to the AMLUnitTest project as documented below.

The android Mobile Control Application (aMCA) is also a good resource for example code. The aMCA\_Manifest contains additional permissions and parameters to more fully exercise library code.

General

* + Android version support is limited to 4.0.x – 4.4.4
  + Items marked TBD may be implemented in a future release
  + Items marked private are for system use only, not supported and subject to change w/o notice
  + API level documentation is in aAyla Mobile Library.docx
  + Methods not documented in aAyla Mobile Library.docx are not supported and subject to change w/o notice
  + All date types defined as time are currently UTC date strings
  + Best to read the documentation all the way through before starting
  + Leverage the unit test code and demo app for working examples of every library call!
  + To step through the code, set a break point at amlTestSettings()

Unit test - AMLUnitTest.java

* Reference the unit test examples in AMLUnitTestActivity.java for simple working code examples
* The unit test program requires a functional standard hardware demo kit (EVB)
* You'll need an Ayla developer account. Create it using Ayla Control or using the developer site <https://developer.aylanetworks.com>.
* Change the values between BEGIN/END UNIT TEST CONFIGURATION to match your situation:
* Change userName and password to the ones from your developer account
* Change gblAmlDsn = "AC000W00000####" to match your hardware developer kit Device Serial Number
* With the EVB connected to the cloud & registered to your developer account, AMLUnitTest should pass
* Change appId, and appSecret to the values supplied by your Ayla support contact. Use the supplied values until then
* Change gblAmlCountryCode & gblAmlPhoneNumber to your SMS enabled phone country code and phone number respectively. Change gblScheduleName to “sched1” if working with a smart plug, else leave it as “schedule\_in” for an EVB.
* Leave gblAmlDeviceSsidRegex alone for now. Later change it to match your products unique SSID AP signature
* Change gblAmlModuleSsid with the SSID of the Demo Kit/EVB in AP mode
* Change gblAmlLanSsid & gblAmlLanSsidPassword to match your WLAN AP with internet access
* Leave amlTestSuite equal to DEVICE\_SERVICE\_ONLY for now. Changing the testSuite allows you to explore specific areas of functionality.
* Be sure to associate the "Ayla\_LED\_EVB demo" template to your EVB/Demo Kit hardware. When testing schedule (disabled by default) use the template “Ayla\_LED\_EVB\_demo\_schedules” and that your uHost build supports schedules.
* Launch the app and tap “Tap here to begin” at the top of the display.

Known issues

SystemUtils

* Configuration settings are bootstrap settings used during initial launch and should not be changed.
* Be sure to call loadSavedSettings() before making any other calls!
* The log file base name is called aml\_log and is located in the Environemnt.getExternalStorageDirectory() (typically /sdcard/). When a file reaches 512KB, it is versioned to .1-5 and then recycled.

LAN Mode Enabled (LME)

* LAN Mode Support allows for direct communication between a host phone/tablet and an EVB
* It is recommended to start with a basic Ayla service oriented application, and then LAN mode enable it.
* Important LME differences and prerequisites
  + The device MUST have “LAN Support” enabled on the Ayla field service (developer.aylanetworks.com)
  + The application MUST enable LAN mode support as described in this section
  + The application MUST have communicated with the Ayla field service at least once
  + Knowledge of the device property names & baseTypes is required prior to retrieving properties from the device.  Therefore, the application MUST retrieve the properties for the device from the Ayla Device Service before properties (values & data points) can be retrieved from a LME device with an active session. Doing so will save the properties as defined in the template to cache. Cached values can be used in future instantiations to derive the property names and base types.
  + The getProperties() method supports the callParams option. This feature allows for specifying the property names of a subset of properties to retrieve. These callParams are ignored for calls made to the Ayla field service and all properties are retrieved.
  + iAML will ignore requests for properties of the base types specified in AML\_LANMODE\_IGNORE\_BASETYPES Aylanetworks.h
  + String baseTypes are limited to 1023 characters
  + Getting more than one datapoint will cause the request to be satisfied by the Ayla field service

User

* Device Service login
  + Be sure to call AylaSystemUtils.init() before calling login.
  + Credentials supported include email/password, Oauth for Google & Facebook
  + Be sure to login before accessing device service calls.
* LAN Mode Login (no Ayla cloud reachability)
  + Security is up to the application developer. Common issues to address include:
  + Checking username and password against the last known good login.
  + Clearing the cache of known devices if a new user logs in to the service.

Devices

The current device must be LAN mode enabled (LME)for direct communication.

Properties

* callParams can be used on LME devices to reduce the number of properties returned

Datapoints

* Getting more than one datapoint on an LME device will cause the request to be routed to the Ayla field service.
* Limit value queries to the current property or the most current datapoint to ensure direct communication.

Property Triggers and Application Triggers

CRUD of property triggers & application triggers is not supported in LAN login mode

Device Notifications and Application Notifications

CRUD of device notification & application notifications is not supported in LAN login mode

User Shares and Device Shares

CRUD of user & device sharing is not supported in LAN login mode

Schedules and Schedule Actions

Schedules require specific support on the EVB for the host MCU. The EVB developer must support both schedules and GPIO mode. It is also mandatory the mobile application use the same schedule names as the host MCU. The defaults for the supplied example code on both the host MCU and aMCA are sched1, sched2, sched3, sched4, and sched5 for oem\_model type of smartplug1. If oem\_model equals ledevb, the schedule in the template and host MCU is named schedule\_in.

Read the AML Schedules documentation before implementing schedules. Then pick your model: Full or Dynamic Action. Next check the device OEM template create the schedule(s) at <https://developer.aylanetworks.com> by selecting the target device->Edit Device->Schedules, if they don’t already exist. Recommend setting the active flag to false initially. If using the Dynamic Action model, also create Schedule Action(s) for each Schedule. Recommend setting the active flag to false initially for Schedule Actions, too.

CRUD of schedules and actions is not supported in LAN login mode

Setup Task

The setup task may be performed without signing-in to the device service. However, registration and the device services require user sign-in. . Only asynchronous interfaces are supported for all setup tasks. It is recommended that users sign-in before doing wifi-setup so that issue logging can be directly associated with a user’s account.

Registration

Registration modes define the requirements for how a newly Setup device is associated with a registered user.

* + Same-LAN Registration
  + The local Ayla device has completed setup and connected to the Ayla device service within the last hour
  + The local Ayla device and the phone/pad/pod/tablet running this code are connected to the same WLAN
  + AP-Mode Registration
  + The local Ayla device has completed setup and connected to the Ayla device service within the last hour
  + The mobile application that completed setup for the new device, is the app used to register the device
  + Push-Button Registration
  + The local Ayla device has completed setup and connected to the Ayla device service.
  + The local Ayla device and the phone/pad/pod/tablet running this code are connected to the same WLAN
  + The registration button on the new device was pushed within the last 2 minutes.
  + Display Registration
  + No same-LAN requirement
  + No registration time window requirement
  + Dsn Registration
  + The local Ayla device has completed setup and connected to the Ayla device service within the last hour
  + No same-LAN requirement

See the Ayla Mobile Library documentation for more registration mode specifics

Debugging

* Most calls are asynchronous so be sure to set a break point at both the public method call and it's associated handler.
* WiFi debugging in the emulator is not supported (WiFi Setup will not work)
* When using the emulator for GUI development, be sure to enable SD Card support and allocate at least 100 MiB of storage in AVD.

* AMLUnitTest
  + If you get an error similar to:

11-03 09:34:49.299: I/com.aylanetworks.aaml.AylaNetworks(9826): F, amlTest, gblAmlTestDevice:null, getDevices - no device assigned

you need to change gblAmlProductName to match your EVB/Demo kit

* + If you get an error similar to:

11-03 09:19:42.969: I/com.aylanetworks.aaml.AylaNetworks(9756): F, amlTest, gblAmlTestProperty:null, getProperties - no property assigned

you need to associate the “Demo” template to your EVB/Demo kit.

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Author | Change Description |  |
| 3.49 | 12/02/14 | D. Myers | * Fix scanForAp error return path to AylaModule * Add received shares (grants) support * Remove country code leading zeros in AylaApplicationTrigger * Add serviceReachability tiemout to info log message * URI support for China AWS services * Add phone\_country\_code to AylaUser class * Add setSupportEmailAddress to AylaSetemsUtils.java * Set reachability timeout even if lanMode is disabled * Add the propety active to AylaPropertyTrigger class |  | |
| 3.33 | 06/20/14 | D. Myers | * Added AylaGrant class to AylaDevice class * Added amOwner() method to AylaDevice class * Removed callParams parameter from AylaDevice getShares() pass through method * Added AylaDeviceNotification & AylaAppNotification classes to aAML & AMLUnitTest |  | |
| 3.30 | 04/25/14 | D. Myers | * New AylaShare class for AylaUser & AylaDevice pass-through methods * Add registration type DSN mode * resendConfirmation() add params appId & appSecret * resetPassword() add params appId & appSecret * Add nickname member to AylaApplicationTrigger class |  | |
| 3.23 | 04/04/14 | D. Myers | * Metadata support for AylaUser & AylaDevice objects & testing * AylaOAuth support methods for GOOG & FB * Added blob support methods & testing * Added long string support & testing * Added propertyTrigger & TriggerApp update() methods & tests * AMLUnitTest improvements * Updated documentation |  | |
| 3.16 | 12/13/13 | D. Myers | * Set location during WiFi Setup * Timezone support * Email logs to Ayla Support * App specific DNS based device service base URL support |  | |
| 3.10 | 11/04/13 | D. Myers | * Added synchronous schedules test * propertyNickname improvement |  | |
| 3.00 | 09/03/13 | D. Myers | * LAN Mode rearchitecture * Centralized caching in new AylaCache class |  | |
| 2.22 | 08/01/13 | D. Myers | * Schedules update |  | |
| 2.20 | 07/12/13 | D. Myers | * Schedule support for Full Template and Dynamic Action models * Slow Connection Settings * New logging service with WiFi Setup support * Push Button registration support |  | |
| 2.00 | N/A | D. Myers | * LAN Mode Support * User Sign-up and account management * AP-Mode registration |  | |
| 1.55\_ENG | N/A | D. Myers | * Settings support for logging enable/disable to the sign-in options menu |  | |
| 1.53 | N/A | D. Myers | * Integrated 1.32 aAML Setup changes * Integrated 1.42\_ENG aAML LAN Mode changes |  | |
| 1.3.2 | N/A | D. Myers | * Optimization for host network connect: Check to see if host is already connected |  | |
| 1.3.1 | N/A | D. Myers | * Added WiFi retry * Added HTTP retries * Bug fixes & optimization |  | |
| 1.3.0 | N/A | D. Myers | * Sets module time, yielding faster setup times * Defaults to 15 retries for confirmation of setup by the service * Improved clean-up of remembered Wi-Fi profiles on entry & exit |  | |
| 1.2.0 | N/A | D. Myers | * Relaxed initial connection restrictions for Setup * Enabled wireless data connectivity * Attempts connection to the same WLAN new device connects if initially connected to new device, or not connected to WLAN |  | |
| 1.1.2 | N/A | D. Myers | * Improved amlUnitTest app * Support for testing: just setup & registration, just device service calls, both * Support for tap to start or auto start on launch * Minor changes to support aMCA v1.0 |  | |
| 1.1.1 | N/A | D. Myers | * Added returnToMainActivity() to Module.connectNewDeviceToService handler for error conditions |  | |
| 1.1.00 | N/A | D. Myers | * Moved to jar packaging * Updated release notes |  | |
| 1.0.10 | N/A | D. Myers | * Removed package import dependency in AylaHostWifiApi.java * Improved documentation |  | |
| 1.0.00 | N/A | D. Myers | * Added Setup Task functionality * Changed package name to com.aylanetworks.aaml * Moved version number to 1.0.00 |  | |
| 0.5.30 | N/A | D. Myers | * Production/staging * Basic registration * Static environment metrics * Email messages |  | |